Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1. (Currently amended) A method for allocating space among a plurality of queues in a buffer, comprising:

sorting ~~all~~ the plurality of queues of the buffer according to size, thereby to establish a sorted order of the queues;

selecting at least one group of the queues ~~consisting of~~comprising a given number of the queues in accordance with said sorted order;

allocating a portion of the space in the buffer to said at least one group, responsive to the number of the queues in said at least one group; and

accepting a data packet into one of the queues in said at least one group responsive to whether the data packet will cause the space occupied in the buffer by the queues in the group to exceed the allocated portion of the space.

2. (Original) A method according to claim 1, wherein selecting said at least one group comprises selecting for inclusion in said at least one group the queues that are largest among said plurality of queues.

3. (Original) A method according to claim 2, wherein allocating the portion of the space comprises setting a size of the space proportional to a sum of a harmonic series.

4. (Currently Amended) A method according to claim 3, wherein setting the size of the space comprises establishing a total maximum buffer size B_k for the k largest output queues, wherein B_k is substantially given by

$$\cancel{Bk}\,\underline{B_k} = \frac{M}{(\ln N+1)}\sum_{i=1}^{k}\frac{1}{i}$$

wherein M is the total buffer space available to all queues, N is the number of output queues, and i is an integer index.

5. (Original) A method according to claim 4, wherein selecting the number k of the queues comprises selecting a fixed number of the queues.

6. (Original) A method according to claim 4, wherein k has a variable value.

7. (Currently amended) A method according to claim 6, wherein k and ~~Bn~~ <u>B_k</u> are set in accordance with the equation:

$$\cancel{Bk}\underline{B_k} = (n + \frac{M}{c^{n+1}}) \bullet \frac{M}{\mathrm{logc}N}$$

wherein n, c and m are parameters such that $1 < c < N$, and n is the largest integer such that m and k satisfy the conditions

$$0 \leq m \leq c^{n+1}$$

and

$$k = \sum_{i=1}^{n} ci + m.$$

8. (Original) A method according to claim 2, wherein buffer space is allocated within said at least one group of queues in accordance with a set of conditions which functionally define a known buffer management policy.

9. (Original) A method according to claim 8, wherein said known buffer management policy is chosen from a group consisting of complete partitioning (CP), sharing with maximum queue lengths (SMXQ), sharing with minimum allocation (SMA), sharing with maximum queue length and minimum allocation (SMQMA), and dynamic threshold (DT).

10. (Original) A method according to claim 2, wherein said given number is fixed.

11. (Currently amended) A ~~buffer management policy~~method according to claim 2, wherein said given number and said portion of the space in the buffer are variable.

12. (Currently Amended) A method for allocating space among N output queues in a buffer of size M, which comprises:

sorting ~~all~~ the plurality of queues of the buffer according to size, thereby to establish a sorted order of the queues;

selecting a number k of said N output queues in accordance with said sorted order;

establishing a total maximum buffer space of ~~Bk~~ B_k for said number k of said N output queues, wherein $B_k < M$;

ascertaining whether acceptance of an arriving packet destined for one of said number k of said N output queues will cause the space in the buffer used by said k queues together to exceed B_k; and

if the acceptance of an arriving packet destined for one of said k queues will cause the space used by said k queues together to exceed B_k, rejecting said packet.

13. (Original) A method according to claim 12, wherein said selecting a number k of said N output queues comprises selecting a number k of said N output queues that are largest among said N output queues.

14. (Currently Amended) A method according to claim 13, wherein establishing the total maximum buffer space comprises setting the maximum buffer size such that ~~Bk~~ B_k is substantially given by

$$\cancel{Bk}\underline{B_k} = \frac{M}{(\ln N+1)}\sum_{i=1}^{k}\frac{1}{i}$$

wherein M is the total buffer space available to all queues, and N is the number of output queues, and *i* is an integer index which refers to an ordinal ranking of the queue size.

15. (Original) A method according to claim 13, wherein the buffer space is allocated within said number k of said N output queues in accordance with a set of conditions which functionally define a known buffer management policy.

16. (Original) A method according to claim 15, wherein said known buffer management policy is chosen from a group consisting of complete partitioning (CP), sharing with maximum queue lengths (SMXQ), sharing with minimum allocation (SMA), sharing with maximum queue length and minimum allocation (SMQMA), and dynamic threshold (DT).

17. (Original) A method according to claim 12, wherein selecting the number k of the queues comprises selecting a fixed number of the queues.

18. (Original) A method according to claim 12, wherein the value of k is variable.

19. (Currently Amended) A method according to claim 18, wherein the values of k and ~~B_n-~~$\underline{B_k}$ are set in accordance with the equation

$$\cancel{Bk}\underline{B_k} = (n + \frac{M}{c^{n+1}}) \bullet \frac{M}{\log cN}$$

wherein n, c and m are parameters such that $1 < c < N$, and n is the largest integer such that in and k satisfy the conditions:

$$0 \leq m \leq c^{n+1}$$

and

$$k = \sum_{i=1}^{n} ci + m.$$

20. (Currently amended) A shared memory switch comprising:

a memory ~~providing~~ configured to provide a buffer space ~~of size M, which~~ that is adapted to contain a plurality of output queues; and

a controller, ~~coupled~~ configured to:

sort all the queues of the buffer according to size, thereby to establish a sorted order of the queues, ~~and~~

~~to~~ allocate the space in the buffer to the output queues in accordance with said sorted order such that a portion of the space is allocated to a group of the output queues ~~consisting of~~comprising a given number of the output queues ~~that are largest among the plurality of the queues~~ responsive to the given number of the output queues in the group, and

accept a data packet into one of the output queues in the group responsive to whether the data packet will cause the space occupied in the buffer by the output queues in the group to exceed the allocated portion of the space.

21. (Original) A switch according to claim 20, wherein said portion of the space is allocated by setting a size of the space proportional to a sum of a harmonic series.

22. (Currently Amended) A switch according to claim 21, wherein said portion of the space is allocated for the k largest output queues and the total maximum buffer size B_k for said k largest output queues is substantially given by

$$\cancel{Bk}\underline{B_k} = \frac{M}{(\ln N+1)}\sum_{i=1}^{k}\frac{1}{i}$$

wherein *i* is an integer index which refers to an to ordinal ranking of the queue size.

23. (Currently Amended) A switch according to claim 22, wherein said portion of the space is allocated for the k largest output queues and the total maximum buffer size $\underline{B_k}$~~Bk~~ for said k largest output queues is substantially given by the equation

$$\cancel{Bk}\underline{B_k} = (n + \frac{M}{c^{n+1}}) \bullet \frac{M}{\log cN}$$

wherein n, c and m are parameters such that $1 < c < N$, and n is the largest integer such that m and k satisfy the conditions

$$0 \leq m \leq c^{n+1}$$

and

$$k = \sum_{i=1}^{n} ci + m.$$

24. (Currently Amended) A ~~shared memory~~ switch according to claim 20, wherein said switch has N output lines and N corresponding output queues, wherein said given number is denoted by k and said portion of the space available for the k largest output queues is denoted by B_k, wherein $B_k < M$.

25. (Currently Amended) A ~~shared memory~~ switch according to claim 24, wherein $\underline{B_k}$~~Bk~~ is substantially given by the equation

$$\cancel{Bk}\underline{B_k} = \frac{M}{(\ln N+1)}\sum_{i=1}^{k}\frac{1}{i}$$

wherein i is an integer index which refers to the ordinal ranking of the queue in terms of the queue size.

26. (New) A device for allocating space among a plurality of queues in a buffer, comprising:

logic configured to sort the plurality of queues of the buffer according to size, thereby to establish a sorted order of the queues;

logic configured to select at least one group of the queues comprising a given number of the queues in accordance with said sorted order;

logic configured to allocate a portion of the space in the buffer to said at least one group, responsive to the number of the queues in said at least one group; and

logic configured to accept a data packet into one of the queues in said at least one group responsive to whether the data packet will cause the space occupied in the buffer by the queues in the group to exceed the allocated portion of the space.

27. (New) The device according to claim 26, wherein the logic configured to select said at least one group comprises logic configured to select for inclusion in said at least one group the queues that are largest among said plurality of queues.

28. (New) The device according to claim 27, wherein the logic configured to allocate the portion of the space comprises setting a size of the space proportional to a sum of a harmonic series.

29. (New) The device according to claim 28, wherein the logic configured to set the size of the space comprises logic configured to establish a total maximum buffer size B_k for the k largest output queues, wherein B_k is substantially given by

$$B_k = \frac{M}{(\ln N + 1)} \sum_{i=1}^{k} \frac{1}{i}$$

wherein M is the total buffer space available to all queues, N is the number of output queues, and *i* is an integer index.

30. (New) The device according to claim 29, wherein selecting the number k of the queues comprises selecting a fixed number of the queues.

31. (New) The device according to claim 29, wherein k has a variable value.

32. (New) The device according to claim 31, wherein k and B_k are set in accordance with the equation:

$$B_k = (n + \frac{M}{c^{n+1}}) \bullet \frac{M}{\log cN}$$

wherein n, c and m are parameters such that $1 < c < N$, and n is the largest integer such that m and k satisfy the conditions

$$0 \leq m \leq c^{n+1}$$

and

$$k = \sum_{i=1}^{n} ci + m.$$

33. (New) The device according to claim 27, wherein buffer space is allocated within said at least one group of queues in accordance with a set of conditions which functionally define a known buffer management policy.

34. (New) The device according to claim 33, wherein said known buffer management policy is chosen from a group consisting of complete partitioning (CP), sharing with maximum queue lengths (SMXQ), sharing with minimum allocation (SMA), sharing with maximum queue length and minimum allocation (SMQMA), and dynamic threshold (DT).

35. (New) The device according to claim 27, wherein said given number is fixed.

36. (New) The device according to claim 27, wherein said given number and said portion of the space in the buffer are variable.

37. (New) A method for allocating space among a plurality of queues in a buffer, comprising:

selecting at least one group of the queues comprising a number of queues in the plurality of queues based on a sorting of the plurality of queues;

allocating a portion of the space in the buffer to said at least one group based on the number of the queues in said at least one group; and

accepting a data packet into one of the queues in said at least one group responsive to whether the data packet will cause the space occupied in the buffer by the queues in the group to exceed the allocated portion of the space.

38. (New) A method according to claim 37, wherein selecting said at least one group comprises selecting for inclusion in said at least one group the queues that are largest among said plurality of queues.

39. (New) A method according to claim 38, wherein allocating the portion of the space comprises setting a size of the space proportional to a sum of a harmonic series.

40. (New) A method according to claim 39, wherein setting the size of the space comprises establishing a total maximum buffer size B_k for the k largest output queues, wherein B_k is substantially given by

$$B_k = \frac{M}{(\ln N + 1)} \sum_{i=1}^{k} \frac{1}{i}$$

wherein M is the total buffer space available to all queues, N is the number of output queues, and *i* is an integer index.

41. (New) A method according to claim 40, wherein selecting the number k of the queues comprises selecting a fixed number of the queues.

42. (New) A method according to claim 40, wherein k has a variable value.

43. (New) A method according to claim 42, wherein k and B_k are set in accordance with the equation:

$$B_k = (n + \frac{M}{c^{n+1}}) \bullet \frac{M}{\log cN}$$

wherein n, c and m are parameters such that $1 < c < N$, and n is the largest integer such that m and k satisfy the conditions

$$0 \le m \le c^{n+1}$$

and

$$k = \sum_{i=1}^{n} ci + m.$$

44. (New) A method according to claim 38, wherein buffer space is allocated within said at least one group of queues in accordance with a set of conditions which functionally define a known buffer management policy.

45. (New) A method according to claim 44, wherein said known buffer management policy is chosen from a group consisting of complete partitioning (CP), sharing

with maximum queue lengths (SMXQ), sharing with minimum allocation (SMA), sharing with maximum queue length and minimum allocation (SMQMA), and dynamic threshold (DT).

46. (New) A method according to claim 38, wherein said given number is fixed.

47. (New) A method according to claim 38, wherein said given number and said portion of the space in the buffer are variable.

48. (New) The method of claim 3, wherein setting the size of the space comprises establishing a total buffer size based on a number, N, of the plurality of queues; a number, k, of the queues in the at least one group; and a ranked order of the queues in the at least one group based on queue length.

49. (New) The method of claim 6, wherein the variable value of k is determined based on a parametric harmonic policy.

50. (New) The method of claim 14, wherein setting the size of the space comprises establishing a total buffer size based on a number, N, of the plurality of queues; a number, k, of the queues in the at least one group; and a ranked order of the queues in the at least one group based on queue length.

51. (New) The method of claim 18, wherein the variable value of k is determined based on a parametric harmonic policy.

52. (New) The device of claim 21, wherein setting the size of the space comprises establishing a total buffer size based on a number, N, of the plurality of queues; a number, k, of the queues in the at least one group; and a ranked order of the queues in the at least one group based on queue length.

53. (New) The device of claim 52, wherein k is variable and the variable value of k is determined based on a parametric harmonic policy.

54. (New) The device of claim 24, wherein setting the size of the space comprises establishing a total buffer size based on a number, N, of the plurality of queues; a number, k, of the queues in the at least one group; and a ranked order of the queues in the at least one group based on queue length.

55. (New) The method of claim 28, wherein setting the size of the space comprises establishing a total buffer size based on a number, N, of the plurality of queues; a number, k, of the queues in the at least one group; and a ranked order of the queues in the at least one group based on queue length.

56. (New) The device of claim 31, wherein the variable value of k is determined based on a parametric harmonic policy.

57. (New) The method of claim 39, wherein setting the size of the space comprises establishing a total buffer size based on a number, N, of the plurality of queues; a number, k, of the queues in the at least one group; and a ranked order of the queues in the at least one group based on queue length.

58. (New) The method of claim 42, wherein the variable value of k is determined based on a parametric harmonic policy.